TITLE OF THE INVENTION

OPTIMUM DESIGN METHOD AND APPARATUS, AND PROGRAM FOR THE SAME

BACKGROUND OF THE INVENTION

Field of the Invention

[0001] The present invention relates to a solution of an optimization problem of design parameters, and more particularly to an automatic design technique for optimization of shapes including topology of structural members.

Description of the Related Art

[0002] The term "structural topology optimum design" means a problem for deciding the topology and shape dimensions of a structural member, which are optimum under given conditions. Hereinafter, the topology and shape dimensions of a structural member will be referred to as "design argument functions", and the above decision problem will be referred to as a "design-variable-function optimization problem". The reason why the term "argument functions" is used resides in that the topology and shape dimensions are given as functions of a three-dimensional space.

[0003] In the design-variable-function optimization problem, an optimization problem of a state argument function must be solved for a value of each design argument function. From this point of view, the structural topology optimum design can be regarded as a double structure optimization problem that contains an optimization problem of a state argument function on the inner side and an optimization problem of a design argument function on the outer side. As to the optimization problem of a state argument function on the inner side, the concept of dividing a space into a finite number of elements is employed based on accumulation of known techniques.

[0004] For a problem taking strain energy of a structural member as an evaluation generic function, in particular, a finite element method is generally employed as a solution technique for that problem. A direct method with respect to a linear equation is employed for solution of the finite element method.

[0005] On the other hand, regarding the optimization problem of a design argument function, the following three kinds of methods have been primarily provided (see Reference 1, S. Bulman, J. Sienz, E. Hinton: "Comparisons between algorithms for structural topology optimization using a series of benchmark studies", Computers and Structures, 79, pp.1203-1218 (2001), or Reference 2, Y-L. Hsu, M-S. Hsu, C-T.

Chen: "Interpreting results from topology optimization using density contours", Computers and Structures, 79, pp.1049-1058 (2001)):

1. Evolutionary method (referred to as "E method" below)
2. Homogenization method (referred to as "H method" below)
3. Material distribution method (referred to as "MD method" below)

[0006] According to the E method, each of partial spaces resulting from dividing a space is called a cell, and creation and disappearance of cells are repeated in accordance with proper rules. A structural member is given as a set of cells that exist finally. A definite structural member is obtained by allowing only two states represented by whether the cell exists or not. Also, because the E method does not employ differential information of the evaluation generic function and can avoid trapping to a local optimum solution, the E method is effective in the case of the evaluation generic function having multiple peaks.

[0007] Reference 3 (Japanese Patent Laid-Open No. 2001-134628) proposes an optimization design device for a skeleton structural member, which employs one kind of the E method, i.e., a genetic search method. The proposed optimization design device comprises an approximate optimization computing device using approximation formulae for discrete design variable data such as section dimensions of a skeleton structural member, and a close optimization computing device using the design variable data. Those two computing devices are combined with each other to be adapted for an actual design problem containing a large number of design variables.

[0008] The H method enables sensitivity analysis to be employed by assuming a finer structure for a structural element positioned in each of divided partial spaces and introducing a new design argument function having a continuous value. Here, the term "sensitivity analysis" means an analytical technique utilizing differential information of the evaluation generic function with respect to the design argument function. If the sensitivity analysis is enabled, it becomes possible to employ an iterative method, such as a gradient method. As a result, a computation time required for search of at least a local optimum solution can be greatly cut in comparison with a round-robin technique, such as the E method. (See Reference 4, Hiroshi Yamakawa, "*Saitekika Dezain* (Optimization Design)", Computation Dynamics and CAE Series 9, Baifukan Co., Ltd. (1996).)

[0009] The MD method is a method for expressing changes in topology and shape dimensions of a structural member by assigning, to each element, a real number in the range of 0 to 1 representing the presence probability of the structural member. The MD method is similar to the H method in that the sensitivity analysis is enabled by replacing discrete information, which indicates whether the structural member is present or not, with a continuous value representing the presence probability of the structural member. However, because the number of parameters is smaller in the MD method than in the H method, the MD method is advantageous in that modeling is easier and the application range is wider.

[0010] Reference 5, Fujii, Suzuki and Otsubo, "Structure Phase Optimization Using Voxel Finite Element Method", Transactions of JSCES, Paper No. 200000010(2000), describes a phase optimization technique for a structure based on the MD method. This technique has features given below.

[0011] (1) Because of using a voxel finite element method (in which a space is divided at equal intervals), element rigidity matrices for all elements are the same. Accordingly, by computing the element rigidity matrix once in advance, it is available in subsequent computations. Furthermore, since elements are regularly arranged, there is no need of storing node number information for each element.

[0012] (2) A pre-processed conjugate gradient method and an element-by-element method are combined with each other to solve large-scaled simultaneous linear equations. As a result, a solution can be obtained without setting up an entire rigidity matrix, and therefore the memory capacity required for processing can be reduced.

[0013] (3) A homogenization method requires 6 design variables (in the three-dimensional case) for one element. Further, each time the design variable is changed, the element rigidity matrix must be computed again. On the other hand, by employing a density method expressing the presence probability of the structural member as a density ratio, only one design variable is required for one element. In addition, changes of the design variable do not affect the element rigidity matrix.

[0014] The above-mentioned known techniques, however, have problems given below.

[0015] Generally, a structure optimization problem is formulated as a double optimization problem containing an optimization problem of a state variable vector per iterative step for an optimization problem of a design variable vector. Assuming the optimization problem of a state variable vector to be optimization on the outer side and the optimization problem of a design variable vector to be optimization on the inner side, the optimization on the inner side is a problem of determining the state variable vector with the design variable vector being a parameter, i.e., the design variable vector being fixed. This problem is usually called structural analysis and can be solved based on the finite element method by using the solution technique for a linear equation.

[0016] However, if a structure changes and a structural member does not exist in a certain area any more, for example, if a structural member is holed, the design variable of the corresponding element becomes 0 and the Young's modulus of that element also eventually becomes 0. This leads to a result that the problem cannot be solved by the direct method because a coefficient matrix of the linear equation is not fully filled with elements and an inverse matrix does not exist.

[0017] For that reason, most of the known techniques employ a method of using a value of not exactly 0, i.e., replacing 0 with a small value close to 0, when the design variable vector becomes 0. This method is similarly applied to the voxel finite element method employed in Reference 5.

[0018] Setting the design variable of the element to a small value, however, corresponds to the case in which there exists a thin film or a weak member from a physical point of view. Stated another way, an area where no materials are present is not exactly expressed in the known techniques.

[0019] Further, as seen from the figure showing the result in Reference 5, a structural element area not contributing to total strain energy, i.e., a floating island area or a projection area, is left in th result. Such a structural element area should be finally removed, but a method for removing that area is not explained in Reference 5.

[0020] In Fig. 16, by way of example, when elements 512, 513 and 514 are neither subjected to any weight nor supported, those elements have no contributions from the standpoint of strength. In other words, those elements are ones not contributing to total strain energy and hence should be removed in accordance with a design demand for minimum total weight of the structural members. However, it is difficult to find those elements so long as the solution technique is based only on simple geometrical information, i.e., on connection of characteristic functions.

SUMMARY OF THE INVENTION

[0021] It is an object of the present invention to provide an optimum design method capable of executing structure optimum design, including topology, without utilizing the experiential rules that depend on individual problems.

[0022] Another object of the present invention is to provide an optimum design method capable of eliminating a component corresponding to a structural element that does to contribute to an evaluation function, when a variable for optimizing a value of the evaluation function is determined.

[0023] According to one aspect, the present invention which achieves those objects relates to an optimum design method comprising a first solution determining unit of solving an optimization problem of a first evaluation function for a state variable vector with a design variable vector being as a parameter; and a second solution determining unit of solving an optimization problem of a second evaluation function for the design variable vector and the state variable vector obtained in the first solution determining unit, wherein the second solution determining unit includes a gradient vector computing unit of computing a gradient vector of the second evaluation function for the design variable vector; a first coefficient computing unit of computing a first coefficient based on a value of a norm of the gradient vector; a search vector computing unit of computing a search vector based on the first coefficient; a second coefficient computing unit of computing a second coefficient; and a design variable vector updating unit of updating the design variable vector based on the second coefficient, the second coefficient computing unit including the first solution determining unit, the first solution determining unit being executed as an iterative method based on the gradient vector, and the state variable vector being not initialized during iteration.

[0024] According to one aspect, the present invention which achieves those objects relates to an optimum design apparatus comprising first solution determining means for solving an optimization problem of a first evaluation function for a state variable vector with a design variable vector being as a parameter; and second solution determining means for solving an optimization problem of a second evaluation function for the design variable vector and the state variable vector obtained in the first solution determining means, wherein the second solution determining means includes gradient vector computing means for computing a gradient vector of the second evaluation function for the design variable vector; first coefficient computing means for computing a first coefficient based on a value of a norm of the gradient vector; search vector computing means for computing a search vector based on the first coefficient; second coefficient computing means for computing a second coefficient; and design variable vector updating means for updating the design variable vector based on the second coefficient, the second coefficient computing means including the first solution determining means, the first solution determining means being executed as an iterative method based on the gradient vector, and the state variable vector being not initialized during iteration.

[0025] According to still another aspect, the present invention which achieves those objects relates to a computer-readable optimum design program, the program comprising codes for causing a computer to perform an optimum design method comprising a first solution determining step of solving an optimization problem of a first evaluation function for a state variable vector with a design variable vector being as a parameter; and a second solution determining step of solving an optimization problem of a second evaluation function for the design variable vector and the state variable vector obtained in the first solution determining step, wherein the second solution determining step includes a gradient vector computing step of computing a gradient vector of the second evaluation function for the design variable vector; a first coefficient computing step of computing a first coefficient based on a value of a norm of the gradient vector; a search vector computing step of computing a search vector based on the first coefficient; a second coefficient computing step of computing a second coefficient; and a design variable vector updating step of updating the design variable vector based on the second coefficient, the second coefficient computing step including the first solution determining step, the first solution determining step being executed as an iterative method based on the gradient vector, and the state variable vector being not initialized during iteration.

[0026] According to yet another aspect, the present invention which achieves those objects relates to an optimum design method comprising a first solution determining step of solving an optimization problem of a first evaluation function for a state variable vector with a design variable vector being as a parameter; and a second solution determining step of solving an optimization problem of a second evaluation function for the design variable vector and the state variable vector obtained in the first solution determining step, wherein the second solution determining step includes a design variable vector updating step of updating the design variable vector in sequence, the design variable vector updating step including a minimum point searching step of making search from a start point to obtain a minimum point; and a terminal point evaluating step of deciding an optimum point based on a value of the second evaluation function at the minimum point and a value of the second evaluation function at an end point.

[0027] According to another aspect, the present invention which achieves those objects relates to an optimum design apparatus comprising first solution determining means for solving an optimization problem of a first evaluation function for a state variable vector with a design variable vector being as a parameter; and second solution determining means for solving an optimization problem of a second evaluation function for the design variable vector and the state variable vector obtained in the first solution determining means, wherein the second solution determining means includes design variable vector updating means for updating the design variable vector in sequence, the design variable vector updating means including minimum point searching means for making search from a start point to obtain a minimum point; and terminal point evaluating means for deciding an optimum point based on a value of the second evaluation function at the minimum point and a value of the second evaluation function at an end point.

[0028] According to a further aspect, the present invention which achieves those objects relates to a computer-readable optimum design program, the program comprising codes for causing a computer to perform an optimum design method comprising a first solution determining step of solving an optimization problem of a first evaluation function for a state variable vector with a design variable vector being as a parameter; and a second solution determining step of solving an optimization problem of a second evaluation function for the design variable vector and the state variable vector obtained in the first solution determining step, wherein the second solution determining step includes a design variable vector updating step of updating the design variable vector in sequence, the design variable vector updating step including a minimum point searching step of making search from a start point to obtain a minimum point; and a terminal point evaluating step of deciding an optimum point based on a value of the second evaluation function at the minimum point and a value of the second evaluation function at an end point.

[0029] According to a further aspect, the present invention which achieves those objects relates to an optimum design method comprising a first solution determining step of solving an optimization problem of a first evaluation function for a state variable vector with a design variable vector being as a parameter; a second solution determining step of solving an optimization problem of a second evaluation function for the design variable vector and the state variable vector obtained in the first solution determining step; and an erasing step of erasing, from the design variable vector, a component corresponding to a structural element that does not contributes to the second evaluation function.

[0030] According to a further aspect, the present invention which achieves those objects relates to an optimum design apparatus comprising first solution determining means for solving an optimization problem of a first evaluation function for a state variable vector with a design variable vector being as a parameter; second solution determining means for solving an optimization problem of a second evaluation function for the design variable vector and the state variable vector obtained in the first solution determining means; and erasing means for erasing, from the design variable vector, a component corresponding to a structural element that does not contributes to the second evaluation function.

[0031] According to a further aspect, the present invention which achieves these objects relates to a computer-readable optimum design program, the program comprising codes for causing a computer to perform an optimum design method comprising a first solution determining step of solving an optimization problem of a first evaluation function for a state variable vector with a design variable vector being as a parameter; a second solution determining step of solving an optimization problem of a second evaluation function for the design variable vector and the state variable vector obtained in the first solution determining step; and an erasing step of erasing, from the design variable vector, a component corresponding to a structural element that does not contributes to the second evaluation function.

[0032] Other objectives and advantages besides those discussed above shall be apparent to those skilled in the art from the description of a preferred embodiment of the invention which follows. In the description, reference is made to accompanying drawings, which form a part thereof, and which illustrate an example of the invention. Such example, however, is not exhaustive of the various embodiments of the invention, and therefore reference is made to the claims which follow the description for determining the scope of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0033] Fig. 1 is a flowchart showing a processing sequence of a second solution determining step.

[0034] Fig. 2 is a block diagram of an apparatus according to one embodiment.

[0035] Fig. 3 is a flowchart showing a processing sequence of a line searching step.

[0036] Fig. 4 is a flowchart showing a processing sequence of a search range narrowing step.

[0037] Fig. 5 is a flowchart showing a processing sequence of a minimum point searching step.

[0038] Fig. 6 is a flowchart showing a processing sequence of a first vector correcting step.

[0039] Fig. 7 is a flowchart showing a processing sequence of a second vector correcting step.

[0040] Fig. 8 is a flowchart showing a processing sequence of a coefficient B computing step.

[0041] Fig. 9 is a flowchart showing a processing sequence of a normal vector computing step.

[0042] Fig. 10 is a flowchart showing a processing sequence of a first solution determining step.

[0043] Fig. 11 is an illustration for explaining practical setting of a problem.

[0044] Fig. 12 is an illustration showing a calculation result in the embodiment.

[0045] Fig. 13 is a flowchart showing a processing sequence for removing an undesired element.

[0046] Fig. 14 is a flowchart showing another processing sequence for removing an undesired element.

[0047] Fig. 15 is an illustration showing a calculation result after removing the undesired element.

[0048] Fig. 16 is an illustration for explaining undesired elements.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0049] A preferable one embodiment according to the present invention will be described in detail below with reference to the accompanying drawings.

(First Embodiment)

[0050] Fig. 2 shows the configuration of an apparatus according to one embodiment. As shown in Fig. 2, the apparatus comprises a CPU 201, a display 202, an input unit 203, a primary storage 204, a secondary storage 205, a communication unit 206, and a bus line 207.

[0051] Various processing functions executed in this embodiment is stored as programs in the secondary storage 205 beforehand. Any of the stored programs is loaded into the primary storage 204 and then executed in response to a command input via the input unit 203 or the communication unit 206.

[0052] For the sake of explanation given below, a target problem is formulated as follows.

[0053] Assuming that an argument function is expressed by a finite dimension vector with formulation based on the finite element method, an evaluation generic function for an argument function is given as an evaluation function for a variable vector. The following description is made on an assumption that the argument function is expressed by a finite dimension vector.

[0054] A state variable vector x and a design variable vector f are written as column vectors, respectively, as given below;

(Eq. 1) $x = (x_1, x_2, \ldots, x_m)^T$

(Eq. 2) $f = (f_1, f_2, \ldots, f_n)^T$

wherein T represents transposition, x represents an m-dimension vector, and f represents an n-dimension vector.

[0055] Boundary conditions of x and f are assumed to be B_1 and B_2, respectively, as follows:

(Eq. 3) $B_1(x) = 0$

(Eq. 4) $B_2(f) = 0$

[0056] Evaluation functions for the state variable vector x and the design variable vector f are assumed to be L_1 and L_2, respectively, as follows;

(Eq. 5) $L_1 = L_1(x;f)$

(Eq. 6) $L_2 = L_2(f,x)$

wherein L_1 is a function with x being a variable vector and f being a parameter, and L_2 is a function with x and f being both variable vectors.

[0057] In a general optimization problem, a number K_1 of equality restraint conditions and a number K_2 of inequality restraint conditions are applied in many cases. Therefore, the j-th equality restraint condition and the k-th inequality restraint condition for the design variable are assumed to be Q_j and R_k, respectively, as follows:

(Eq. 7) $Q_j(f,x) = 0$

(Eq. 8) $R_k(f,x) \geq 0$

[0058] From the definition described above, the optimum design problem is given as a problem of determining a solution, which satisfies the following equation, under the restraint conditions (Eq. 4), (Eq. 5), (Eq. 6), (Eq. 7) and

(Eq. 8):

(Eq. 9) $\min[L_2]$

[0059] Additionally, the state variable x is obtained as a solution satisfying the following equation under the restraint condition (Eq. 3):

(Eq. 10) $\min[L_1]$

[0060] Processing executed in this embodiment will be described below on the basis of the above formulation.

[0061] For simplicity of the explanation, it is assumed in the following description that the equality restraint conditions of (Eq. 7) is given by one condition as follows;

(Eq. 11) $$Q(f) = \sum_{j=1}^{j=n} C_j f_t(j) + c_0 = 0$$

wherein c_j ($j = 0, 1, \ldots, n$) is a constant.

[0062] Also, the inequality restraint conditions of (Eq. 8) are assumed to be given as the following ranges of value;

(Eq. 12) $$R_j^-(f) = f_t(j) - f_{j\min} \geq 0$$
$$R_j^+(f) = f_{j\max} - f_t(j) \geq 0$$

wherein $j = 1, 2, \ldots, n$.

[0063] Accordingly, the number of the inequality restraint conditions is twice the number of the design variables.

[0064] Flag vectors a^+ and a^- having the same dimensions as those of the design variable vector are defined as follows:

$a^+ = 0$, if $0 \geq R_j^+(f)$

$a^+ = 1$, otherwise

and

$a^- = 0$, if $0 \geq R_j^-(f)$

$a^- = 1$, otherwise

[0065] Also, it is assumed that a set of j, at which a^+ and a^- are each 1, is Λ_1 and a set of other indices is Λ_2.

[0066] Fig. 1 is a flowchart showing a processing sequence of a second solution determining step. Referring to Fig. 1, in step S101, the CPU reads various factors of a system to be simulated. The various factors may be read as data entered from the input unit 203 or the communication unit 206, or may be given by reading data that has been stored as a file in the secondary storage 205 beforehand. The various factors of the system include initial values of x and f, the boundary conditions B_1, B_2, the evaluation functions L_1, L_2, and the restraint conditions Q_j, R_k. Based on such information of the various factors, the program secures a required variable area in the primary storage 204 and sets values.

[0067] In step S102, the number of times t of iterations is initialized to 1. The number of times t of iterations means the number of times at which a first iteration process has been executed in the second solution determining step. The second solution determining step is brought to an end if the number of times of iterations of the processing from step S103 to step S111 reaches a setting value T.

[0068] A gradient vector computing step is executed in step S103. In the gradient vector computing step, the CPU computes a gradient vector g_t of a second evaluation function for the design variable vector. The gradient vector g_t is computed in accordance with the following equation:

(Eq. 13) $g_t = (\partial L_2/\partial f_1, \partial L_2/\partial_2, \ldots, \partial L_2/\partial f_n)^T$

[0069] Whether $\partial L_2/\partial f_j$ can be analytically solved or not depends on cases. If an analytical solution is present, (Eq. 10) is obtained by utilizing a computing process for the analytical solution. If otherwise, the gradient vector can be computed by utilizing the technique called automatic differentiation. The automatic differentiation technique is known from, for example, Reference 6 (Koichi Kubota and Masao Iri, "*Arugorizumu no Jidobibun to Oyou* ("Automatic Differentiation and Applications of Algorithm", Modern Nonlinear Science Series, Corona Publishing Co., Ltd., (1998)).

[0070] In step S104, a first vector correcting step is executed for g_t obtained from (Eq. 13). The first vector correcting step will be described later with reference to Fig. 6.

[0071] A convergence determining step is executed in step S105. In this step, the square of a norm of g_t is computed, and if the computed value becomes 0 or smaller than a very small value ε close to 0, the processing is brought to an end. The square of the norm is computed in accordance with the following equation;

(Eq. 14) $\|g_t\|^2 = g_t{}^T g_t$

[0072] A first coefficient computing step is executed in step S106. In the first coefficient computing step, a coefficient β is computed in accordance with (Eq. 15) given below:

(Eq. 15) $\beta = \|g_t\|^2 / \|g_{t-1}\|^2$

wherein $\beta = 1$ holds at $t = 1$.

[0073] A search vector computing step is executed in step S107. In this step, a search vector p_t at the t-th iteration is computed in accordance with (Eq. 16) given below:

(Eq. 16) $p_t \leftarrow -g_{t-1} + \beta p_{t-1}$

[0074] In step S108, a first vector correcting step is executed for p_t obtained from (Eq. 16). This step will be described later with reference to Fig. 6.

[0075] A line searching step is executed in step S109. In this step, a coefficient δ for updating the design variable is decided. This step will be described later with reference to Fig. 3.

[0076] A design variable vector updating step is executed in step S110. In this step, the design variable vector is updated as follows:

(Eq. 17) $f_{t+1} = f_t + \delta p_t$

[0077] In step S111, a second vector correcting step is executed for the design variable vector. This step will be described later with reference to Fig. 7.

[0078] In step S112, t is updated to t+1, and if t after the update exceeds the preset number of times of iterations, the processing is brought to an end.

[0079] The foregoing is the flow of the second solution determining step. Individual processing steps in this step will be described in more detail below.

[0080] The line searching step will be described below with reference to Fig. 3. In this step, the current design variable value f_t and the search vector p_t at the current design variable value are given as arguments. It is also assumed that the evaluation function and the restraint condition are given as appropriate.

[0081] In step S301, a maximum step size Δ is computed using the following equation;

(Eq. 18) $$\Delta = \underset{j \in \Lambda_1}{MIN} [(ff(j) - f_t(j)) / p_t(j)]$$

wherein $ff(j) = f_{jmax}$, if $p_t(j) > 0$

$= f_{jmin}$, if $p_t(j) < 0$

[0082] Further, various variables are initialized as follows;

(Eq. 19) $g1 = (3.0 - 5.0^{1/2}) / 2$

$$g2 = (5.0^{1/2} - 1) / 2$$

$$d\Delta = \Delta/T_m$$

$$a1 = 0.0$$

$$a2 = a1 + d\Delta$$

wherein T_m is a positive integer used for dividing the search range into appropriate zones and is set to a value in the range of, e.g., 10 to 100.

[0083] In step S302, a search range narrowing step is executed. This step will be described later with reference to Fig. 4.

[0084] In step S303, δ set in the search range narrowing step is compared with the maximum step size Δ. If they are equal to each other, the processing is brought to an end. If otherwise, the CPU proceeds to step S304.

[0085] In step S304, a minimum point searching step is executed. This step will be described later with reference to Fig. 5. The line searching step is thereby brought an end.

[0086] The search range narrowing step will be described below with reference to Fig. 4.

[0087] In step S401, values of the second evaluation functions are computed as follows;

(Eq. 20) $L21 = L2(x(f_t + a1p_t), f_t + a1p_t)$

(Eq. 21) $L22 = L2(x(f_t + a2p_t), f_t + a2p_t)$

wherein $x(f_t + a1p_t)$ and $x(f_t + a2p_t)$ are obtained by executing a first solution determining step that will be described later with reference to Fig. 10.

[0088] In step S402, L21 and L22 are compared with each other. If L21 is larger than L22, the processing is brought to an end. If otherwise, the CPU proceeds to step S403.

[0089] In step S403, t is initialized to 0 and $\mu(t)$ is initialized to 0.

[0090] In step S404, $\mu(t)$ is updated as follows:

(Eq. 22) $\mu(t) = \mu(t-1) + d\Delta$

[0091] In step S405, L21 and L22 are updated as follows;

(Eq. 23) $L21 = L22$

(Eq. 24) $L22 = L_2(x(f_x + \mu(t)p_t), f_x + \mu(t)p_t)$

wherein $x(f_x + \mu(t)p_t)$ is obtained by executing the first solution determining step that will be described later with reference to Fig. 12.

[0092] In step S406, L21 and L22 are compared with each other. If L21 is not larger than L22, the CPU proceeds to step S408, and if otherwise, it proceeds to step S407.

[0093] In step S407, t is updated to t+1, and thereafter t is compared with T_m. If t is larger than T_m, the CPU proceeds to step S409, and if otherwise, it proceeds to step S404.

[0094] In step S408, computations of the following equations are executed:

(Eq. 25) $a1 = \mu(t) - 2d\Delta$

$a2 = \mu(t)$

[0095] In step S409, $\mu(t)$ and Δ are compared with each other. If $\mu(t)$ is larger than Δ, Δ is substituted for δ. If otherwise, an arbitrary value other than Δ is substituted for δ, whereby the search range narrowing step is brought to an end.

[0096] The minimum point searching step will be described below with reference to Fig. 5.

[0097] In step S501, a computation of the following equation is executed:

(Eq. 26) $da = a2 - a1$

[0098] If da is not larger than a preset small positive real number, the CPU proceeds to step S504, and if otherwise, it proceeds to step S502.

[0099] In step S502, computations of the following equations are executed:

(Eq. 27) $v1 = a1 + g1\ da$

$v2 = a1 + g2\ da$

[0100] Further, the search range is narrowed as follows:

(Eq. 28) $a2 = v2$, if $L21 < L22$

$a1 = v1$, otherwise

[0101] This narrowing is a range reduction based on the golden section method.

[0102] In step S503, values of the second evaluation function are computed based on the following equations, and the CPU proceeds to step S501;

(Eq. 29) $L21 = L2(x(f_x + a1\ p_t),\ f_x + a1\ p_t)$

$L22 = L2(x(f_x + a2\ p_t),\ f_x + a2\ p_t)$

wherein $x(f_x + a1p_t)$ and $x(f_x + a2p_t)$ are obtained by executing the first solution determining step that will be described later with reference to Fig. 10.

[0103] In step S504, a middle point ac of the search zone and a value L2c of the second evaluation function at the middle point are computed as follows:

(Eq. 30) $ac = (a1 + a2)/2$

(Eq. 31) $L2c = L2(x(f_x + ac\ p_t),\ f_x + ac\ p_t)$

[0104] In step S505, a value of δ is decided as follows:

(Eq. 32) $\delta = ac$, if $L2c = MIN[L21,L22,L2c,L2e]$

$= a1$, else if $L21 = MIN[L21,L22,L2c,L2e]$

$= a2$, else if $L22 = MIN[L21,L22,L2c,L2e]$

$= \Delta$, else if $L2e = MIN[L21,L22,L2c,L2e]$

$= 0$, otherwise

[0105] In (Eq. 32), MIN[] is a function of returning a minimum value among the arguments. The minimum point searching step is thereby brought to an end.

[0106] The first vector correcting step executed in each of step S104 and step S108 will be described below with reference to Fig. 6.

[0107] A second coefficient computing step is executed in step S601. In this step, values of a unit normal vector NN, a length DST of a vertical line from the origin to a hyperplane, and a coefficient B are computed. The second coefficient computing step will be described in detail later with reference to Fig. 8.

[0108] A first vector projecting step is executed in step S602. In this step, a vector X is corrected in accordance with an equation given below by using the unit normal vector NN, the length DST of the vertical line from the origin to the hyperplane, and the coefficient B, which have been computed in step S601:

$$\text{(Eq. 33)} \quad X(j) = X(j) - B\,NN(j),\ \text{if } j \in \Lambda_1$$
$$= 0, \qquad \text{otherwise}$$

[0109] The first vector correcting step is thereby brought to an end.

[0110] The second vector correcting step executed in step S111 will be described with reference to Fig. 7.

[0111] The second coefficient computing step is executed in step S701. This step will be described in detail later with reference to Fig. 8.

[0112] A second vector projecting step is executed in step S702. In this step, the vector X is corrected in accordance with an equation given below by using the unit normal vector NN, the length DST of the vertical line from the origin to the hyperplane, and the coefficient B, which have been computed in step S701:

(Eq. 34) $$\begin{aligned} X(j) &= X(j) - B\ NN(j), && \text{if } j \in \Lambda_1 \\ &= f_{jmax}, && \text{if } a^+ = 0 \\ &= f_{jmin}, && \text{if } a^- = 0 \end{aligned}$$

[0113] The second vector correcting step is thereby brought to an end.

[0114] The second coefficient computing step will be described below with reference to Fig. 8.

[0115] In step S801, a normal vector computing step is executed to obtain NN and DST. This step will be described in detail later with reference to Fig. 9.

[0116] In step S802, the coefficient B is computed using the following equation:

(Eq. 35) $$B = \sum_{k \in \Lambda_1} u(j)NN(j) - DST$$

[0117] The second coefficient computing step is thereby brought to an end.

[0118] The normal vector computing step executed in step S801 will be described below with reference to Fig. 9. In this step, the unit normal vector NN and the length DST of the vertical line from the origin to the hyperplane are computed.

[0119] In step S901, the variables expressed by the following equations are computed;

(Eq. 36) $$DST = -c(0)'$$

wherein c(0)' is calculated in accordance with an equation given below;

(Eq. 37) $c(0)' = c(0) + \sum_{k\in\Lambda_2} c(k)f_t(k)$

(Eq. 38) $D = (\sum_{k\in\Lambda_1} c(k)^2)^{-1/2}$

(Eq. 39) $\sum_{j\in\Lambda_1} c(j)f_t(j) + c(0)' = 0$

[0120] DST in Eq. 36 is corrected in accordance with an equation given below;

(Eq. 40) DST = DST / D

[0121] DST given by above (Eq. 40) provides the length of the vertical line from the origin to the hyperplane.

[0122] In step S902, a normal vector computing step is executed.

[0123] First, a normal vector N not normalized is computed in accordance with (Eq. 41) and (Eq. 42) given below:

(Eq. 41) $N(j) = \prod_{k\in\Lambda_1} c(k) / c(j),$ if $j\in \Lambda_1$

(Eq. 42) $N(j) = f_t(j),$ otherwise

[0124] By using (Eq. 41), a unit normal vector NN for a partial space is computed in accordance with an equation given below;

(Eq. 43) $NN(j) = N(j)(\sum_{j\in\Lambda_1} \prod_{k\in\Lambda_1} c(k)^2 / c(i)^2)^{-1/2},$ if $j\in \Lambda_1$

[0125] For example, when the total sum of elements of the design variable vector is 1 as the equality restraint condition, components of the unit normal vector is simplified as follows because of c(0) = -1, c(j) = 1, j = 1, 2,…, n;

(Eq. 44) $NN(j) = |\Lambda_1|^{-1/2}$, if $j \in \Lambda_1$

wherein $|\Lambda_1|$ is the number of elements of Λ_1.

[0126] The normal vector computing step is thereby brought to an end.

[0127] The first solution determining step will be described below with reference to Fig. 10.

[0128] If a structure changes, there is a possibility that an element having a characteristic function value of 0 appears. In the case of carrying out a structural analysis for such a structure, a coefficient matrix becomes not regular and the problem cannot be solved by the direct method. For that reason, one of the following three methods is employed in the first solution determining step:

(1) The characteristic function value being 0 is replaced with a small value close to 0, and then simultaneous linear equations are solved by the direct method.

(2) Simultaneous linear equations are reconstructed using only the elements having values above 0, and then the direct method is employed to solve the problem.

(3) The problem is solved by an iterative method using a

gradient vector, which does not require a computation of an inverse matrix. This iterative method includes a steepest descent method, a conjugate gradient method, etc.

[0129] The first solution determining step using the conjugate gradient method will be described below.

[0130] In step S1001, the CPU reads various factors of a system to be simulated. The various factors include an initial value of x, a value of the design parameter f, the boundary condition B_1, and the evaluation function L_1. Based on such information of the various factors, the program secures a required variable area in the primary storage 204 and sets values.

[0131] In step S1002, the number of times t of iterations is initialized to 1. The number of times t of iterations means the number of times at which a first iteration process has been executed in the first solution determining step. The first solution determining step is brought to an end if the number of times of iterations of the processing from step S1003 to step S1010 reaches a setting value.

[0132] In step S1003, the CPU computes a gradient vector $g_{1,t}$ of a first evaluation function for the state variable vector. The gradient vector $g_{1,t}$ is computed in accordance with the following equation:

(Eq. 45) $g_{1,t} = (\partial L_1/\partial x_1, \partial L_1/\partial x_2, \ldots, \partial L_1/\partial x_n)^T$

[0133] Whether $\partial L_1/\partial x_j$ can be analytically solved or not depends on cases. If an analytical solution is present, (Eq. 45) is obtained by utilizing a computing process for the analytical solution. If otherwise, the gradient vector can be computed by utilizing the technique called automatic differentiation. The automatic differentiation technique is known from, for example, the above-cited Reference 6, etc.

[0134] In step S1004, the square of a norm of $g_{1.t}$ is computed, and if the computed value becomes 0 or a value very close to 0, the processing is brought to an end. The square of the norm is computed in accordance with the following equation;

(Eq. 46) $\|g_{1,t}\|^2 = g_{1,t}{}^T g_{1,t}$

[0135] In step S1005, a coefficient β is computed in accordance with (Eq. 47) given below;

(Eq. 47) $\beta = \|g_{1,t}\|^2 / \|g_{1,t-1}\|^2$

wherein $\beta = 1$ holds at $t = 1$.

[0136] In step S1006, a search direction vector p_t is computed based on (Eq. 48) given below;

(Eq. 48) $p_{1,t} = -g_{1,t-1} + \beta p_{1,t-1}$

[0137] In step S1007, a line searching step is executed. This step is to decide a coefficient α for updating the design variable. When the structural analysis problem is solved by using the finite element model, a value of α is given by the following equation:

(Eq. 49) $\alpha = p_{1,t}{}^T g_{1,t} / (p_{1,t}{}^T A p_{1,t})$

[0138] In step S108, the state variable vector is updated as follows:

(Eq. 50) $x_{t+1} = x_t + \alpha\ p_{1,t}$

[0139] In step S1009, t is updated to t+1, and if t after the update exceeds the preset number of times of iterations, the processing is brought to an end.

[0140] The first solution determining step is thereby brought to an end.

(Practical Example)

[0141] In this practical example, the present invention is applied to optimum shape automatic design for a cantilevered beam that is subjected to a weight at an arbitrary position. For the sake of simplicity, the following description is made on an assumption being limited to a problem of plane strain.

[0142] As shown in Fig. 11, a design area in which a structural member is able to exist is a rectangle 1102. According to the finite element method, the design area is divided at equal intervals into numbers n_y, n_x of partial areas, respectively, in the vertical (length) and horizontal (width) directions. The divided partial areas are called cells and numbered such that a lower left cell and an upper right cell are expressed respectively by (1,1), (n_y,n_x). Lattices points are called nodes and likewise numbered such that a lower left node and an upper right node are expressed respectively by (1,1), (n_y+1,n_x+1).

[0143] In Fig. 11, numeral 1101 denotes a support member, and 1103 denotes a weight vector. Each cell (j,k) has a corresponding characteristic function value f(j,k). Here, the term "characteristic function value" means a variable taking a positive real number value in the range of 0 to 1, which represents the existence probability of the structural member in the cell (j,k), and it is an element of the design variable vector f, given below, in the present invention:

(Eq. 51) $f = (f(1,1), f(1,2),\dots, f(n_y,n_x))^T$

[0144] Similarly, each node (j,k) has a corresponding horizontal displacement u(j,k) and vertical displacement v(j,k). These displacements are real numbers taking arbitrary values and are elements of the state variable vector U, given below, in the present invention:

(Eq. 52) $U = (u(1,1), v(1,1), u(1,2), v(1,2),\dots,$
$u(n_y+1,n_x+1), v(n_y+1,n_x+1))^T$

[0145] Assuming a rigidity matrix to be A and a weight vector to be b, as a well-known result based on the finite element method, the state variable vector U is given as a solution of the optimum problem of an evaluation generic function expressed by the following equation:

(Eq. 53) $L_1 = (1/2)\ U^TAU - b^TU$

[0146] More specifically, it is known that the state variable vector U is given as a solution of a linear equation given below:

(Eq. 54) $AU - b = 0$

[0147] Further, $g_{1,t}$ computed in the first solution determining step is given as the left side of (Eq. 54). Because (Eq. 54) is a set of simultaneous linear equations, it is usually solved by the direct method based on an inverse matrix. In the structure optimum problem, however, the rigidity matrix A is a function of the design variable vector f. Therefore, if f contains an element taking a value of 0, the rigidity matrix A is not fully filled with elements and becomes not regular. This results in that the simultaneous linear equations cannot be solved by the direct method. For that reason, the present invention employs the conjugate gradient method.

[0148] An evaluation function L_2 for the design variable vector f is defined by total strain energy as follows:

(Eq. 55) $L_2 = (1/2)U^T AU$

[0149] Then, g_t required in the second solution determining step can be computed in accordance with the following equation;

(Eq. 56) $g_t = \partial L_2/\partial f(e) = -(1/2)\ U_e{}^T A_e U_e$

wherein U_e, A_e are respectively a displacement vector at a node corresponding to an element e and an element rigidity matrix corresponding to the displacement vector.

[0150] Fig. 12 shows a computation result obtained in this example. In Fig. 12, a black area represents an area in which the structural member exists.

[0151] In this example, the aspect ratio of the design area for the structural member is 2 : 1 (length/width) and its analytical solution is, as known, a combined structure of beams arranged at ± 45 degrees relative to the horizontal direction.

[0152] It is understood that the computation result shown in Fig. 12 is well matched with the known analytical solution.

[0153] Thus, according to the embodiment described above, the structure optimum design including topology can be executed without utilizing the experiential rules that depend on individual problems.

[0154] An undesired element removing process will be described below. Fig. 13 is a flowchart showing a processing sequence for removing an undesired element. Referring to Fig. 13, in step S1301, the CPU reads various factors of a system to be simulated. The various factors may be read as data entered from the input unit 203 or the communication unit 206, or may be given by reading data that has been stored as a file in the secondary storage 205 beforehand. The various factors of the system include initial values of x and f, the boundary conditions B_1, B_2, the evaluation functions L_1, L_2, and the restraint conditions

Q_j, R_k. Based on such information of the various factors, the program secures a required variable area in the primary storage 204 and sets values.

[0155] In step S1302, a solution of the optimization problem under the inequality restraint conditions formulated as (Eq. 4) to (Eq. 9) is determined. While the above-described method can be used in this solution determining step, the undesired element removing process described below is also applicable to the case of employing any of the following methods:

(1) sequential linear plan method

(2) feasible direction method

(3) gradient projection method

(4) general contraction gradient method

(5) optimum reference method

[0156] Subsequent steps S1303 to S1308 constitute the undesired element removing process.

[0157] First, in step S1303, an element *s* is initialized to 1.

[0158] In step S1304, it is checked whether f(s) is 0. If so, the CPU proceeds to step S1308, and if otherwise, it proceeds to step S1305.

[0159] In step S1305, the value of f(s) is changed as follows:

(Eq. 57) $f(s) \leftarrow f(s) + \delta$

wherein δ is an arbitrary real number other than 0, at which f(s) takes a value not smaller than 0, but not larger than 1.

[0160] In step S1306, sensitivity $\partial L_2/\partial f(s)$ of a second evaluation function for f(s) is computed as follows:

$$\partial L_2/\partial f = \partial/\partial f[(1/2)U^T AU]$$

[0161] From a rigidity equation of AU = b, the following equation is obtained:

$$\text{(Eq. 58)} \quad \partial L_2/\partial f = \partial/\partial f[(1/2)b^T U] = (1/2)b^T \partial U/\partial f$$

[0162] On the other hand, assuming the weight b to be constant without depending on f, partial differentiation of both sides of the rigidity equation of AU = b by f leads to:

$$\text{(Eq. 59)} \quad \partial U/\partial f = -A^{-1}\partial A/\partial f U$$

because of $(\partial A/\partial f)U + A(\partial U/\partial f) = 0$.

[0163] By putting (Eq. 59) in (Eq. 58),

$$\text{(Eq. 60)} \quad \partial L_2/\partial f = -(1/2)U^T(\partial A/\partial f)U$$

is obtained from the relation of $U = A^{-1}b$.

[0164] The above equation can be rewritten for a characteristic function of the element *s* as follows;

$$\text{(Eq. 61)} \quad \partial L_2/\partial f(s) = -(1/2)U_s^T(\partial A_s/\partial f(s))U_s$$

wherein U_s is a vector constituted by a displacement on the node belonging to the element *s*, and A_s is an element rigidity matrix corresponding to U_s.

[0165] Whether $\partial A_e/\partial f(e)$ can be analytically solved or not depends on cases. If an analytical solution is present, a computing process for the analytical solution can be obtained. In the case of $A_s = f(s)A_s'$, for example, $\partial L_2/\partial f(s)$ is given by the following equation.

(Eq. 62) $\partial L_2/\partial f(s) = -(1/2)U_s^T A_s' U_s$

[0166] If $\partial A/\partial f(s)$ cannot be analytically solved, it is possible to execute the computation by utilizing the above-mentioned technique called automatic differentiation.

[0167] Here, the sensitivity expressed by (Eq. 62) is given as 0 or a small value close to 0 for all the elements as a result of the solution determining step executed in step S1302.

[0168] A description will be made of whether the value of (Eq. 62) is changed or not when the characteristic function value f(s) of the element *s* is changed in accordance with (Eq. 57).

[0169] First, when the element *s* contributes to the value of strain energy, i.e., when $(1/2)U_s^T A_s U_s$ is not 0, A_s changes with a change of f(s), and therefore the strain energy $(1/2)U_s^T A_s U_s$ of the element *s* also changes. Thus, U_s obtained as the solution of the structural analysis problem takes different values U_s' between before and after the change of f(s). Accordingly, (Eq. 62) takes a negative value as follows:

(Eq. 63) $\partial L_2/\partial f'(s) = -(1/2)U_s'^T A_s' U_s' \neq -(1/2)U_s^T A_s' U_s = 0$

[0170] On the other hand, when the element *s* does not contribute to the value of strain energy, i.e., when

$(1/2)U_s^TA_sU_s$ is 0, U_s becomes 0 because of A_s being a positive definite matrix, and therefore $(1/2)U_s^TA_sU_s$ does not change even with a change of f(s). Accordingly, U_s obtained as the solution of the structural analysis problem takes the same value between before and after the change of f(s), whereby (Eq. 62) becomes 0.

[0171] Stated another way, whether the relevant structural element is necessary or not can be determined by checking whether an absolute value of the sensitivity resulting upon a change in the value of f(s) is 0 or a small value close to 0.

[0172] In step S1307, whether the sensitivity $\partial L_2/\partial f(s)$ is 0 is checked. If so, f(s) is updated to 0, and if otherwise, f(s) is not updated.

[0173] In step S1308, *s* is updated to s+1. Then, if updated *s* exceeds n, the processing is brought an end, and if otherwise, the CPU proceeds to step S1304.

[0174] In the method described above, the undesired element removing step is executed as a post-process. However, when the second solution determining step is executed as an iterative process, the undesired element removing step may be executed in the iteration loop. The processing in such a case will be described below with reference to Fig. 14. The conjugate gradient method is employed, by way for example, as the iterative process in the following description.

[0175] In step S1401, the CPU executes initialization. More specifically, the design variable vector f is set to a predetermined value, which written as f_0.

[0176] In step S1402, t is set to 1.

[0177] In step S1403, a value which the gradient vector g_t of the second evaluation function for the design variable vector f takes at $f = f_0$ is computed as follows:

(Eq. 64) $g_t \equiv \partial L_2/\partial f$

$= (\partial L_2/\partial f(1), \partial L_2/\partial f(2), \ldots, \partial L_2/\partial f(n))^T$

[0178] In step S1404, g_t is corrected so as to satisfy the equality restraint condition and the inequality restraint condition.

[0179] In step S1405, it is checked whether a norm computed in accordance with the following equation exceeds a preset value. If the norm exceeds the preset value, the processing is brought to an end, and if otherwise, the CPU proceeds to step S1406;

(Eq. 65) $\|g_t\| = (g_t^T g_t)^{-1/2}$

[0180] In step S1406, β defined by an equation given below is computed:

(Eq. 66) $\beta = \|g_t\| / \|g_{t-1}\|$

wherein $\beta = 0$ holds at $t = 1$.

[0181] In step S1407, the search vector p_t is computed in accordance with the following equation:

(Eq. 67) $p_t = \beta p_t^{-} g_t$

[0182] In step S1408, p_t is corrected so as to satisfy the equality restraint condition and the inequality restraint condition.

[0183] In step S1409, line search is executed along p_t to find f that minimizes the second evaluation function. The found f is set as f_t.

[0184] In step S1410, f_t is corrected so as to satisfy the equality restraint condition and the inequality restraint condition.

[0185] In step S1411, the undesired element removing step is executed.

[0186] In step S1412, t is updated to t+1, and if t after the update exceeds a preset value, the processing is brought to an end. If otherwise, the CPU proceeds to step S1403.

[0187] Additionally, the processing executed in steps S1404, S1408 and S1410 can be made by using a technique described as the gradient projection method in the above-cited Reference 4.

(Practical Example)

[0188] In this practical example, the above-described embodiment is applied to optimum shape automatic design for a cantilevered beam that is subjected to a weight at an arbitrary position. For the sake of simplicity, the following description is made on an assumption being limited to a problem of plane strain.

[0189] The evaluation function L_2 for the design variable vector f is defined by total strain energy in accordance with (Eq. 55) mentioned above:

(Eq. 55) $L_2 = (1/2)U^T AU$

[0190] Then, f minimizing (Eq. 55) is determined. This kind of problem is usually subjected to, as the equality restraint condition, a restriction on the total weight being constant, namely;

(Eq. 68) $$\sum_{j=1}^{j=n_y} \sum_{k=1}^{k=n_x} f(j,k) = \text{constant}$$

and to, as the inequality restraint condition, a restriction on the value range which the characteristic function value can take, namely:

(Eq. 69) $0 \leq f(j,k) \leq 1$

[0191] As described above, that optimization problem under the inequality restraint condition can be solved by the known solution determining method.

[0192] The undesired element removing step in this practical example will be described below.

[0193] In particular, the computation of sensitivity executed in step S1306 of Fig. 13 can be made by using the following equation;

(Eq. 70) $\partial L_2/\partial f(j,k) = -(1/2)U_{j,k}{}^T A_{j,k} U_{j,k}$

wherein $U_{j,k}$ is an element displacement vector having, as a component, a displacement on the node belonging to the element (j,k), and $A_{j,k}$ is an element rigidity matrix corresponding to the element displacement vector.

[0194] Subsequently, a floating island element can be removed in accordance with the method described above with reference to Fig. 13.

[0195] Fig. 15 shows the shape of the structural member after the undesired element removing step has been executed on the shape of the structural member shown in Fig. 12. It can be confirmed that floating island areas and projections appearing in Fig. 12 are eliminated in Fig. 15. In this example, the undesired element is determined and removed only when the absolute value of the sensitivity is exactly equal to 0. A computing time required for the undesired element removing step is about 40 seconds when the CPU is Pentium III (933 MHz).

[0196] Incidentally, the present invention is applicable to not only an apparatus constituted by a single unit, but also to a system constituted by a plurality of units. Further, the present invention may be implemented by supplying, to an apparatus or a system, a storage medium storing program codes of software for realizing the functions of the above-described embodiment, and by causing a computer in the apparatus or the system to read and execute the program codes stored in the storage medium so that the apparatus or the system achieves the desired functions.

[0197] In addition to the case of directly realizing the functions of the above-described embodiment by the computer in the apparatus or the system executing the program codes stored in the storage medium, the present invention also involves the case of realizing the functions of the above-described embodiment by causing an OS (Operating System), etc. operating on the computer to execute the processing in accordance with instructions of the program codes.

[0198] In such a case, a storage medium storing the program codes constitutes the present invention.

[0199] Although the present invention has been described in its preferred form with a certain degree of particularity, many apparently widely different embodiments of the invention can be made without departing from the spirit and the scope thereof. It is to be understood that the invention is not limited to the specific embodiments thereof except as defined in the appended claims.